首先我们考虑数列中出现的0，只要有0存在，那么数列中所有数字的积必定为0，所以我们先把数列中的所有0变成1.在进行这样的操作后，我们再检查一下这个时候数列中所有数字的和是否为0，如果为0 的话再加一次1就可以了（和为0的话，必定又有负数又有整数，随便选一个正数+1就能避免0的再次出现）。

#include<bits/stdc++.h>

#define INF 0x3f3f3f3f //1061109567

#define llINF 9223372036854775807

#define pi 3.141592653589793

#define ll long long

**using** **namespace** std;

**const** ll maxn=1e2+7;

**const** **double** eps=1e-10;

**const** ll mod=1e9+7;

#define IOS ios::sync\_with\_stdio(0); cin.tie(0); cout.tie(0);

ll num[maxn];

**int** main()

{

    IOS;

    ll t;

    cin>>t;

**while**(t--)

    {

        ll n;

        cin>>n;

        ll sum=0,num0=0;

**for**(ll i=1;i<=n;i++)

        {

            cin>>num[i];

**if**(num[i]==0) num0++;

            sum+=num[i];

        }

        ll ans=num0;

**if**(sum+num0==0) cout<<ans+1<<endl;

**else** cout<<ans<<endl;

    }

}